

Javascript Fade Effects

By [Roger Keays](#), 28 August 2007

Recently, I was asked to build a rotating testimonials block in the sidebar of a site. The block would display each testimonial for 5 - 10 seconds before it changed. Loud and rapid changing elements on a website make me cringe. It's like listening to somebody scrape their fingernails on the blackboard. Conversely, soft or slowly changing elements make me feel great! No, really, they do - especially fade effects. I can watch them over and over fading in and out... it feels so good.

I'd seen some people put non-flash fade effects to good use on a few sites so I thought I'd try to give it a go myself. Nobody wants reams and reams of code for some simple effects like this though, so I implemented a few simple javascript functions to do the job for me. Here's how it works.

The fade effect is implemented by modifying the opacity of the element. Recent versions of Firefox, Safari and Opera implement the CSS3 opacity style which can be used as you might expect. Older versions of the browsers, and Microsoft Internet Explorer have the own proprietary methods to set the opacity. If you wanted to give an element fixed opacity you could use the following CSS declarations:

```
#element {
    opacity: 0.5;                /* CSS3 */
    -moz-opacity: 0.5;          /* legacy Firefox */
    -khtml-opacity: 0.5;        /* legacy Safari/Konqueror */
    filter: alpha(opacity=50); /* MSIE */
}
```

Luckily, none of these styles conflict so we don't need any CSS hacks. To dynamically change the opacity, we can use javascript (the following function assumes you've declared an *element* variable):

```
/* set the opacity of the element (between 0.0 and 1.0) */
function setOpacity(level) {
    element.style.opacity = level;
    element.style.MozOpacity = level;
    element.style.KhtmlOpacity = level;
    element.style.filter = "alpha(opacity=" + (level * 100) + ")";
}
```

Now all that is required is to create timers to change the opacity periodically. A timer is needed for every increment during the fade, so we create them using a loop:

```

var duration = 1000; /* 1000 millisecond fade = 1 sec */
var steps = 20;      /* number of opacity intervals */
var delay = 5000;    /* 5 sec delay before fading out */

function fadeIn(){
  for (i = 0; i <= 1; i += (1 / steps)) {
    setTimeout("setOpacity(" + i + ")", i * duration);
  }
  setTimeout("fadeOut()", delay);
}

function fadeOut() {
  for (i = 0; i <= 1; i += (1 / steps)) {
    setTimeout("setOpacity(" + (1 - i) + ")", i * duration);
  }
  setTimeout("fadeIn()", duration);
}

```

That's pretty much all there is to it, you'll need to call `fadeIn()` to kickstart the effect though. I've made [an example of the whole thing put together](#) so you can see it in action and use it as starting point. The example doesn't rotate the text - it's purely just to demonstrate the fade effects.

[View the example of Javascript Fade Effects.](#)

Credit for these ideas goes to [brainererror.net](#) [1], I've just simplified the work on that site. There are some more complex fade effects on that site which use the same basic principle described here (e.g. fading from image to image).

One other thing, to make this work on IE, the element being faded must be in `hasLayout` mode. The simplest way of doing this is to add the style `zoom: 1;` or give the element a width or height.

References

[1] <http://brainererror.net/scripts/javascript/blendtrans/>

About Roger Keays



Roger Keays is an artist, an engineer, and a student of life. He has no fixed address and has left footprints on 40-something different countries around the world.

Roger is addicted to surfing. His other interests are music, psychology, languages, the proper use of semicolons, and finding good food.